Uso de Aprendizado de Máquina para Categorização Automática de Conjuntos de Dados de Portais de Dados Abertos

Mateus Rangel[1], **Flavia Bernardini**[1], **José Viterbo**[1], **Rodrigo Monteiro**[1], **Elaine Seixas**[1], **Higor dos Santos Pinto**[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF) – Niterói – RJ – Brasil

mateusrangel@id.uff.br, fcbernardini@ic.uff.br, viterbo@ic.uff.br

salvador@ic.uff.br, higorsantos@id.uff.br, elaine_rangel@id.uff.br

Abstract. *To make their data available to society, city governments around the world are using open data portals. In most portals, datasets are arranged into several categories that represent the topics covered by the portal. In this context, providing mechanisms to help categorize datasets becomes important to facilitate the work of an open data portal administrator. In this paper, we present an experimental analysis for automatic categorization of data sets from open data portals using supervised machine learning. In our methodology, we use the dataset name and its attached file attributes to infer its category. For text processing, we used natural language processing techniques.*

Resumo. *Para disponibilizar seus dados para a sociedade, governos de cidades ao redor do mundo estão usando portais de dados abertos. Na maioria dos portais, os conjuntos de dados estão distribuídos por diversas categorias que representam os tópicos abordados pelo portal. Nesse contexto, oferecer mecanismos para auxiliar a categorização dos conjuntos de dados se torna importante, para facilitar o trabalho de um administrador de portais de dados abertos. Neste trabalho, apresentamos uma análise experimental para a categorização automática de conjuntos de dados de portais de dados abertos utilizando aprendizado de máquina supervisionado. Utilizamos o nome do conjunto de dados e os seus atributos de arquivos anexados para a inferência de sua categoria. Para processamento de textos, usamos técnicas de processamento de linguagem natural.*

1. Introdução

Nos últimos anos, governos de cidades ao redor do mundo vêm disponibilizando seus dados de forma aberta, por meio de portais na internet, como uma forma de atender à demanda de transparência da sociedade. Por meio desses portais, a sociedade pode consultar e requisitar bases de dados para obter informações úteis sobre áreas como saúde, transporte, segurança, etc. Os Dados Abertos Governamentais são comumente vistos como incentivadores da eficiência e transparência, da participação do cidadão nas decisões e da inovação na sociedade [Jetzek et al. 2014]. Entretanto, para que o acesso aos dados pelos cidadãos seja efetivo, é necessário que os portais não sejam estruturados como meros repositórios de dados, mas que ofereçam recursos para facilitar a busca por informações [Reis et al. 2018].

Encontrar conjuntos de dados ideais para realizar uma análise, pode ser uma tarefa bem custosa para o usuário. Isso inclui visitar diferentes portais, inspecionar muitos conjuntos de dados e avaliar a qualidade e a relevância dos dados ([Koesten et al. 2017], [Xiao et al. 2019]). Um dos desafios que são encontrados ao fazer alguma análise sobre portais de um conjunto de cidades é a sua integração do ponto de vista das categorias em que os conjuntos de dados estão agrupados. Por não haver um padrão pré-definido, existem casos em que portais diferentes se referem às categorias que representam uma mesma área com nomes diferentes, como, por exemplo, "Public Safety"no portal da cidade de Newark [1] e "Police"no portal de Birmingham [2]. Em outras vezes, uma única categoria em um portal, abrange dados de múltiplas categorias em outros portais, como, por exemplo, "'Infrastructure"no portal de Newark e "Parks/Recreation"e "Transportation"no portal da Filadélfia. Nesse contexto, oferecer mecanismos para auxiliar a categorização é importante, para facilitar o trabalho de um administrador do portal de dados governamentais abertos.

Uma maneira para facilitar a integração de conjuntos de dados, proposta em [dos Santos Pinto et al. 2018], consiste na geração de um subconjunto abrangente de categorias a partir do conjunto de categorias dos portais que se deseja integrar.

O objetivo deste trabalho é apresentar uma análise experimental para avaliar a eficácia do uso de aprendizado de máquina para, dado um conjunto de categorias, categorizar de maneira automática um conjunto de dados a ser disponibilizado em um portal de dados governamentais abertos. Para isso, utilizamos dados coletados de portais de dados governamentais abertos, incluindo conjunto de categorias e metadados dos conjuntos de dados.

Este trabalho está dividido como segue: Na Seção 2 é apresentada a fundamentação teórica e revisão da literatura deste trabalho. Na Seção 3 é apresentada a metodologia da nossa análise experimental. Na Seção 4 são apresentados os resultados obtidos na análise experimental. No Capítulo 5 são apresentadas as conclusões e trabalhos futuros.

2. Fundamentação Teórica e Revisão da Literatura

2.1. Dados Abertos

A Open Knowledge International [Open Knowledge International 20 b] define dados abertos como: "dados e conteúdos que podem ser usados, modificados e compartilhados por qualquer pessoa para qualquer propósito" [Open Knowledge International 20 a]. Suas principais características são [Open Knowledge International 20 c]:

- Disponibilidade: os dados devem estar disponíveis a um custo de reprodução razoável, de preferência através de *download* pela internet. Os dados também devem estar disponíveis de forma conveniente e modificável;
- Reutilização e redistribuição: os dados devem ser fornecidos em termos que permitam a reutilização e redistribuição, incluindo o intercâmbio com outros conjuntos de dados. Os dados devem ser legíveis por máquina;

[1] Disponível em http://data.ci.newark.nj.us/

[2] Disponível em https://data.birminghamal.gov/

- Participação universal: todos devem poder usar, reutilizar e redistribuir, não deve haver discriminação contra os campos de trabalho ou contra pessoas ou grupos.

Para tornar seus dados públicos e, consequentemente, promover a transparência, participação do cidadão e crescimento econômico [Jetzek et al. 2014], governos de cidades ao redor do mundo fazem uso de portais de dados. Esses portais são formados por páginas *web* onde conjuntos de dados sobre várias áreas de atuação do governo da cidade podem ser publicados seguindo a definição de dados abertos descrita anteriormente. As categorias disponíveis em um portal de dados governamentais abertos representam os assuntos que foram abordados em seus conjuntos de dados, como serviços, transporte, planejamento, finanças e saúde. Informações como nome, descrição, licença, mantenedor, data de criação, data de modificação, autor, entre outras, costumam ser encontradas nos arquivos de metadados dos conjuntos de dados de portais de dados abertos [Barbosa et al. 2014].

2.2. Aprendizado de máquina

Aprendizado de máquina [Faceli et al. 2011] é uma área de estudo voltada para que computadores possam reconhecer padrões após aprender com experiência passada. Considerando a abordagem clássica, entre os tipos de aprendizado de máquina temos o aprendizado supervisionado, o aprendizado não supervisionado e o aprendizado por reforço.

O objetivo do aprendizado de máquina supervisionado é construir uma função a partir dos dados de treinamento que possa ser utilizada para prever um rótulo. No nosso caso, como os rótulos pertencem a um conjunto discretos de valores, chamamos o problema de classificação [Faceli et al. 2011].

O aprendizado de máquina não supervisionado, por sua vez, busca explorar recursos ou estruturas ocultas dos dados, baseando-se em dados de entrada não rotulados [Wang et al. 2020]. É considerado não supervisionado por sua capacidade de aprender e organizar informações sem fornecer um sinal de erro para avaliar a solução em potencial [Sathya and Abraham 2013].

Já no aprendizado por reforço, diferente dos demais, não se busca identificar uma categoria como no aprendizado supervisionado, nem encontrar estruturas ocultas como no aprendizado não supervisionado. Neste tipo de aprendizado o treinamento é feito com base em tentativa e erro, aprendendo a partir da interação com o ambiente [Wang et al. 2020].

Os algoritmos de aprendizado de máquina utilizados neste trabalho aceitam como entrada dados em formato atributo-valor. Para que possamos aplicar algoritmos de aprendizado de máquina em textos, é necessário transformar o conteúdo de textos nesse formato, o que implica que cada documento passa a ser representado como um vetor numérico. Nesse processo, é bastante comum a remoção de *stopwords* e estemização dos termos.

As *stopwords* são um conjunto de palavras comuns em um texto em um idioma, e em geral é composto pelas preposições, artigos, dentre outros [Manning et al. 2008]. Já a estemização é um processo para remover as terminações morfológicas e inflexionais mais comuns das palavras, para normalização de termos [Porter 2006]. Neste trabalho, o algoritmo utilizado foi o Porter Stemmer [M.F.Porter 1980] na implementação do NLTK.

O NLTK [NLTK 2019] é uma plataforma para desenvolvimento de programas feita na linguagem de programação Python para tarefas de PLN. Ela disponibiliza um pacote de bibliotecas de processamento textual para classificação, tokenização, *stemming*, etc. Neste trabalho utilizamos a sua implementação do porter stemmer e seu conjunto de *stopwords* em inglês. A seguir descrevemos as técnicas de Processamento de Linguagem Natural (PLN) utilizadas para a vetorização textual [Manning and Schütze 1999], realizada após a remoção de *stopwords* e stemização. Ambas as técnicas também estão implementadas no NLTK. Na sequência, descrevemos brevemente os algoritmos de aprendizado de máquina utilizados neste trabalho, implementados na ferramenta Scikit-learn [Scikit-Learn 2019] é uma biblioteca de aprendizado de máquina para a linguagem de programação Python. Ela inclui, além de algoritmos clássicos de aprendizado de máquina, métodos para todo ciclo de vida do processo de aprendizado de máquina como pré-processamento, redução de dimensionalidade, avaliação/seleção de modelos, dentre outros. Para maiores detalhes dos algoritmos, recomendamos Facelli et al [2011]. Por fim, descrevemos as métricas utilizadas para avaliação dos algoritmos de aprendizado de máquina.

Vetorização Bag of Words: Na abordagem de vetorização *bag-of-words* (saco de palavras), um vocabulário contendo todas as palavras que estiveram presentes em pelo menos um dos exemplos do conjunto de dados é gerado. Em seguida, cada exemplo do conjunto de dados é representado como um vetor que indica a quantidade de vezes que a palavra esteve presente no mesmo. Uma *bag-of-words* é dita binária se representar apenas a presença da palavra em um exemplo [Manning et al. 2008].

Vetorização *Term Frequency–Inverse Document Frequency*(TF-IDF): Em problemas de categorização de textos, tipicamente as categorias são identificadas por palavras que as caracterizam. À primeira vista podemos pensar que as palavras que mais aparecem em documentos pertencentes à uma categoria são as que a definem. Porém, os melhores indicadores de uma categoria são as palavras que mais aparecem naquela categoria e não em outras [Rajaraman and Ullman 2011]. A medida formal de o quão concentrada em relativamente menos documentos são as ocorrências de uma palavra é chamada TF.IDF [Rajaraman and Ullman 2011] sendo uma abreviação do inglês para *Term Frequency times Inverse Document Frequency* que significa frequência do termo multiplicado pelo inverso da frequência nos documentos. Suponha uma coleção de N documentos. Defina f_{ij} a frequência do termo i no documento j. Então, definimos *term frequency* por $TF_{ij} = \frac{f_{ij}}{max_k f_{kj}}$, na qual a frequência do termo i no documento j (f_{ij}) é normalizada pela divisão da maior ocorrência de qualquer termo k no mesmo documento($max_k f_{kj}$). Já o IDF de um termo é definido por $IDF_i = log_2\left(\frac{N}{n_i}\right)$, onde n_i é o número de documentos em que o termo i está presente. Assim, o $TF.IDF$ de um termo i em um documento j é dado por $TF_{ij} \times IDF_i$. Os termos com maior valor de $TF.IDF$ são geralmente os termos que melhor caracterizam o tópico do documento.

Algoritmo de aprendizado Multinomial Naive Bayes: O algoritmo de aprendizado Naive Bayes possui esse nome por assumir que os valores dos atributos de um exemplo são independentes entre si dada a classe. $P(\mathbf{x}|y_i)$ pode ser decomposto no produto $P(x^1|y_i) \times ... \times P(x^d|y_i)$, em que x^j é o j-ésimo atributo do exemplo x. Suponha que queremos classificar um exemplo $\mathbf{x}$ entre duas ou mais classes. Usando o método de estimativa MAP (*Maximum A Posteriori*), considerando cada classe y_i, classificamos o

exemplo $\mathbf{x}$ como sendo da classe que obtiver a maior probabilidade. Já o Multinomial Naive Bayes [Manning et al. 2008] é um modelo de eventos do classificador naive bayes. Modelos de eventos são suposições sobre a distribuição dos atributos. Utilizando o multinomial naive bayes, a probabilidade do j-ésimo atributo do exemplo $\mathbf{x}$ pertencer à classe y_i, $P(x^j|y_i)$ é calculada como a frequência relativa do termo j pertencentes à classe y_i e é dada pela equação 1, onde T_{ct} é o número de ocorrências do termo t em documentos de treino da classe c, e $\sum_{t' \in V} T_{ct'}$ é a contagem do número de ocorrências de todos os termos do vocabulário V na classe c. Repare que estamos adicionando 1 para cada contagem para eliminar zeros. Isso é feito para podermos tratar casos em que estamos calculando a probabilidade de um termo que não estava em nenhum exemplo do conjunto de treinamento, essa técnica se chama *Laplace smoothing*.

$$P(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} \tag{1}$$

Algoritmos de aprendizado para construção de Máquinas de Vetor Suporte: As Máquinas de Vetor Suporte (SVM, do inglês *Support Vector Machine*) são modelos utilizados para classificação baseados em vetores de suporte. Para sua construção, são utilizados algoritmos de aprendizado baseados na teoria de aprendizado estatístico. Inicialmente foram propostos para problemas de classificação binária linear, mas posteriormente o conceito no qual se baseiam esses algoritmos foi ampliado para ser utilizado em problemas de classificação multiclasse, dentre outros. Esses algoritmos resolvem um problema de otimização quadrática para minimização de uma função lagrangiana, cuja solução possui ampla teoria matemática. Uma das desvantagens desses artigos está na diversidade de parâmetros que podem ser utilizados para sua construção, o que também demanda tempo de experimentação para cada domínio de aplicação. Ainda assim, têm sido bastante exploradas pois em muitos problemas no aprendizado de máquina as SVMs têm apresentado bons resultados [Faceli et al. 2011]. Deve ser observado que neste trabalho foram utilizados algoritmos para construção de máquinas de vetor suporte lineares.

Medidas de Desempenho: As medidas utilizadas no cálculo do desempenho do classificador em nosso trabalho são: (i) Acurácia (Acc): proporção de exemplos corretamente classificados; (ii) Precisão ($Prec$): proporção de exemplos de uma classe C classificados corretamente entre todos aqueles preditos como pertencentes à classe C; (iii) Revocação ($Recall$): proporção de exemplos da classe C que foram corretamente preditos; e (iv) F1-Score: Como a precisão não diz quantos exemplos da classe C não foram classificados corretamente e a revocação não diz quantos outros exemplos foram classificados incorretamente como pertencendo à classe C. A precisão e a revocação são combinadas na medida F1-score que consiste na média harmônica da precisão ($prec$) e a revocação ($recall$), dada or $F1 = \frac{2 \times Prec \times Recall}{Prec + Recall}$. Por estarmos em um problema multiclasse, a precisão, revocação e f1-score foram calculadas usando a média macro. Na média macro, a métrica é calculada para cada classe, e em seguida é calculada a sua média aritmética sem peso por classe.

2.3. Revisão da Literatura

Pinto, Bernardini e Viterbo [dos Santos Pinto et al. 2018] apresentaram uma pesquisa exploratória sobre 100 portais de cidades americanas densamente populosas. Nesta pesquisa, mostram como os portais categorizam seus conjunto de dados e sugerem um

método para obter uma categorização genérica para os conjuntos de dados pertencentes aos portais.

Com base nessa pesquisa exploratória, [dos Santos Pinto 2018] apresenta um processo para obtenção do conjunto genérico de categorias, denominado Subconjunto Abrangente, ainda disponibiliza o código fonte do algoritmo, o qual foi utilizado neste trabalho. Abrangente, neste contexto, significa que o conjunto de categorias gerado consegue descrever grande parte dos conjuntos de dados de todos os portais utilizados como entrada. Ainda no mesmo trabalho, os autores apresentam um processo de alinhamento das categorias dos portais com o Subconjunto Abrangente. Neste processo, cada categoria de cada portal é alinhada com uma das categorias do Subconjunto Abrangente utilizando o cálculo de similaridade semântica[Mihalcea et al. 2006]. O cálculo da similaridade semântica entre uma categoria de um portal e uma categoria do Subconjunto Abrangente é feito através do cálculo da similaridade semântica entre todas as palavras que formam as categorias.

Como forma de categorizar conjuntos de dados ainda sem categoria em um portal de dados abertos, [Frtunić Gligorijević et al. 2019] introduzem um classificador chamado EODClassifier framework. Este classificador tem como base a análise formal do conceito como forma de gerar uma estrutura de dados que revela uma conceitualização compartilhada originada do uso de tags.

Pelo fato de portais de dados governamentais abertos não seguirem um padrão de estruturas de categorização, [Yang et al. 2015] tentam avaliar a qualidade da estrutura de categorização de portais de dados abertos automaticamente ao investigar a similaridade dos conjuntos de dados que estão contidos na mesma categoria.

3. Metodologia Utilizada para a Análise Experimental

Na Figura 1 é apresentado o processo metodológico para execução da análise experimental realizada neste trabalho. Na nossa avaliação experimental, a pessoa que pode executar o nosso processo é um administrador de portais de dados abertos, que chamamos de usuário a seguir. A seguir descrevemos cada uma das atividades:

- *Obter os conjuntos de dados*: o usuário deve ter acesso aos conjuntos de dados do portal de dados abertos desejado, contendo: nome, arquivos anexados e categoria. A coleta dos dados pode ser realizada manualmente ou automaticamente;
- *Extrair nome de colunas dos arquivos anexados*: devem ser extraídos os nomes de suas colunas de cada tabela dos dados. A obtenção das colunas vai depender do tipo de arquivo e sua respectiva formatação. Tipos de arquivos bastante comuns em arquivos anexados de portais de dados abertos são CSV(*Comma-separated values*) e GeoJSON. Em arquivos CSV, seus nomes de coluna normalmente se encontram na primeira linha do arquivo sendo separados por vírgula. Em arquivos GeoJSON, seus nomes de coluna são encontrados dentro do objeto FeatureCollection, dentro de objetos contidos no *array features*, sendo as chaves dentro do objeto *properties*;
- *Construir conjunto de dados para o algoritmo de aprendizado de máquina*: deve ser criada uma tupla contendo os campos texto e categoria para cada conjunto de dados coletado. Nesse caso, texto é a concatenação do nome do conjunto de dados com suas colunas extraídas sendo unidos por espaço, e categoria é a categoria do conjunto de dados;

- *Pré-Processar o conteúdo textual*: devem ser realizadas as seguintes etapas: (i) separar por *underline*, pois um dos padrões na nomenclatura de colunas/atributos em bases de dados é a separação de palavras por *underline*. Para separar uma sentença unida por *underline*, basta substituir o *underline* com um espaço em branco; (ii) separar por *camel case*, pois um dos padrões na nomenclatura de colunas em arquivos de dados é a união de palavras por *camel case*. Para separar uma sentença unida por *camel case*, ao detectar a mudança de *casing* em um nome de uma coluna, considere todos os caracteres que vieram antes da mudança como um termo e o restante como outro termo e os una com espaço; (iii) retirar dígitos numéricos, pois números no nome e colunas de um conjunto de dados não costumam adicionar nenhum valor no contexto de uma tarefa de classificação textual. Durante as primeiras iterações do processo de avaliação da metodologia, detectamos valores de anos como *tokens* em nossos vetores textuais, por isso fizemos a remoção; (iv) remover as *stopwords*, que envolve remover também uma lista de palavras que são muito comuns em arquivos de dados abertos e georreferenciados: 'objectid', 'shape', 'length', 'area', 'y', 'x', 'id', 'zip', 'date', 'address', 'code', 'street', 'district', 'lat', 'lng', 'latitude', 'longitude'; e (v) realizar a stemização;
- *Aplicar algoritmos de Vetorização Textual*: são utilizadas as técnicas de Bag of Words e TF-IDF, apresentadas anteriormente;
- *Aplicar algoritmos de aprendizado de máquina*: são utilizados os algoritmos Multinomial Naive Bayes e SVM, descritos anteriormente;
- *Utilizar modelo gerado*, o usuário pode obter uma predição automática de um conjunto de dados ainda sem categoria, seguindo os passos anteriores para obter a representação vetorial do conjunto de dados e então dar como entrada para o classificador gerado.

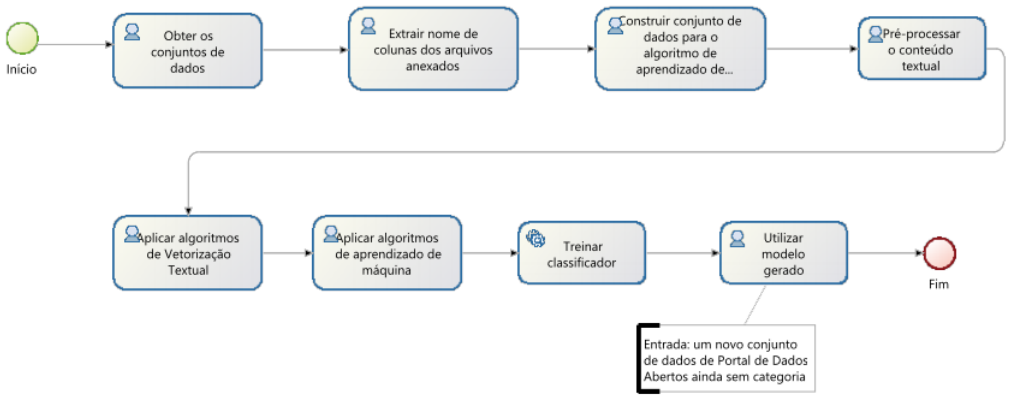

Figura 1. Processo metodológico utilizado para a análise experimental.

Para executar nossa análise experimental, inicialmente tivemos que coletar os conjuntos de dados dos portais. Para a construção de nosso conjunto de dados, foram utilizados 5 portais de cidades americanas dentre as 100 cidades americanas mais populosas abordadas por [dos Santos Pinto et al. 2018], que usam o CKAN como plataforma de por-

Categoria	Frequência Absoluta	Frequência Relativa
Services	56	0.254545
Transportation	49	0.222727
Planning	43	0.195455
Development	32	0.145455
Safety	18	0.081818
Finance	11	0.050000
Health	7	0.031818
Engineering	4	0.018182

Tabela 1. Frequência absoluta e relativa das categorias dos conjuntos de dados

tais de dados abertos: Houston [3], Philadelphia [4], Lexington [5], Newark[6] e Birgmingham[7]. A escolha desses portais também se deu por utilizarem a mesma ferramenta de criação de portais de dados abertos, fazendo com que seja reaproveitável a forma de extração de dados e metadados dos cinco portais. Para que os portais possuam o mesmo conjunto de categorias, executamos a rotina de geração de subconjunto abrangente de categorias dos portais desenvolvida e disponibilizada em [dos Santos Pinto 2018]. Após a execução, 8 categorias foram geradas, sendo elas: *Services*, *Transportation*, *Planning*, *Development*, *Safety*, *Finance*, *Health*, *Engineering*. Neste trabalho, usamos apenas os conjuntos de dados cujas categorias tenham sido usadas para formar o subconjunto de categorias abrangente. Ou seja, categorias cujo nome possua uma das palavras mais abrangentes, exemplo: a categoria 'Public Works & Engineering' faz parte do conjunto de categorias que possui a palavra 'engineering' que foi usada para obter a categoria abrangente 'engineering'. A distribuição de frequências das categorias dos conjuntos de dados utilizados está apresentada na Tabela 1.

A construção do conjunto de dados em formato atributo valor para a execução dos algoritmos de aprendizado de máquina se deu da seguinte forma: Para cada conjunto de dados de cada portal, que é um exemplo de treinamento e teste para o aprendizado de máquina, criamos uma tupla com os atributos texto e *category*. O atributo texto contém o nome do conjunto de dados e o nome dos atributos de seus arquivos anexados e o atributo *category* contém a categoria do conjunto de dados. Apenas arquivos na forma CSV e GeoJSON tiveram seus atributos extraídos. Está disponível em https://github.com/mateusrangel/tcc-resources/blob/master/corpus.csv o arquivo contendo cada exemplo (dados extraídos de cada conjunto de dados de cada portal) com sua respectiva categoria.

Os algoritmos Multinomial Naive Bayes e Máquinas de Vetor Suporte Lineares têm se mostrado técnicas eficientes na categorização de textos na literatura [Colas and Brazdil 2006].

Os parâmetros utilizados para o algoritmo Linear SVM foram: $Penalty$:

[3]Disponível em http://data.houstontx.gov/
[4]Disponível em https://www.opendataphilly.org/
[5]Disponível em https://data.lexingtonky.gov/
[6]Disponível em http://data.ci.newark.nj.us/
[7]Disponível em https://data.birminghamal.gov/

Vetorização	$\overline{Acc}$	$\overline{Prec}$	$\overline{Recall}$	$\overline{F1}$
TF-IDF	0.43	0.26	0.24	0.22
Bag of words	0.46	0.47	0.40	0.41
Bag of Words binária	0.52	0.56	0.45	0.46

Tabela 2. MNB: Resultados das métricas por algoritmo de vetorização

Vetorização	$\overline{Acc}$	$\overline{Prec}$	$\overline{Recall}$	$\overline{F1}$
TF-IDF	0.54	0.62	0.56	0.56
Bag of words	0.52	0.56	0.53	0.51
Bag of Words binária	0.58	0.65	0.60	0.59

Tabela 3. SVM Linear: Resultados das métricas por algoritmo de vetorização

$l2; loss =' squared\_hinge'; dual = True; tol = 1e-4; C = 1.0; multi\_class =' ovr'; fit\_intercept = True; intercept\_scaling = 1; class\_weight = None; verbose = 0; random\_state = None; max\_iter = 1000.$

Os parâmetros utilizados para o Multinomial Naive Bayes foram: $alpha = 1.0; fit\_prior = true; class\_prior = None$

Para mais informações sobre descricao e alternativas de cada parâmetro, ver em scikit-learn LinearSVC[8] e MultinomialNB[9]

Para analisar o desempenho dos modelos construídos, utilizamos a técnica de amostragem *holdout* [Faceli et al. 2011]. No *holdout*, o conjunto de dados gerado na etapa de construção de *dataset* é aleatoriamente dividido em conjunto de treinamento e teste. Em nosso caso, a proporção escolhida foi 2/3 para treinamento e 1/3 para teste. Para fazer com que os resultados obtidos fossem menos dependentes da partição aleatória gerada, aplicamos o particionamento *holdout* 10 vezes e calculamos a média aritmética das medidas de desempenho delas. Desta forma, para cada partição aleatória de treino gerada, os algoritmos de vetorização textual irão gerar um novo vocabulário contendo os termos que estiveram presentes em exemplos da partição treino, e em seguida cada exemplo de treino é representado como um vetor numérico.

4. Resultados da Análise Experimental

Nas Tabelas 2 e 3 são exibidos os resultados obtidos na avaliação realizada utilizando os algoritmos Multinomial Naive Bayes (MNB) e SVM lineares respectivamente. Na primeira coluna de cada tabela são apresentados os métodos de vetorização utilizados; e nas colunas dois a cinco, são apresentadas as medidas de acurácia (Acc), precisão ($Prec$), revocação ($Recall$) e F1-score ($F1$). Podemos observar que a melhor acurácia e F1-score obtidos foram utilizando o algoritmo de aprendizado de máquina Máquinas de Vetores de Suporte Linear com o modelo de vetorização textual *bag of words* binária.

[8]Disponível em https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[9]Disponível em https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

5. Conclusão

Neste trabalho, apresentamos uma análise experimental para a categorização automática de conjuntos de dados de portais de dados governamentais abertos utilizando aprendizado de máquina supervisionado. Em nossa análise, utilizamos o conteúdo textual do nome do conjunto de dados e dos atributos dos arquivos anexados ao mesmo para inferir a categoria a qual ele pertence. Extraímos conjuntos de dados de cinco portais das maiores cidades dos Estados Unidos que utilizam a plataforma CKAN como portal de dados abertos. Os resultados obtidos em nossa análise indicam uma acurácia de classificação de 58% quando utilizando a técnica de vetorização *Bag of Words* binária e o algoritmo de aprendizado Máquinas de Vetores de Suporte Lineares.

Como limitação do nosso trabalho, observamos que a extração de termos se mostra ineficiente em casos em que o atributo de um arquivo de dados possui um nome que consiste de duas ou mais palavras sendo unidas sem nenhum indicador claro, dependendo apenas da interpretação do leitor humano, por exemplo, "numerodefilhos" em que para nós é claro que significa "numero de filhos". Ainda a inferência de uma categoria usou como atributos de entrada apenas o nome do conjunto de dados e o nome das colunas dos seus arquivos anexados. O conteúdo das tuplas/registros dos arquivos de dados também poderiam ser utilizados como entrada para a indução de uma categoria. Por fim, em nosso trabalho, por termos utilizado apenas conjunto de dados cuja categoria estava inclusa no conjunto de categorias que possuíam uma das palavras mais significativas em seu nome, muitos conjuntos de dados foram descartados. Um possível trabalho futuro é, após a geração do classificador desenvolvido em nosso trabalho, inferir categorias do subconjunto abrangente para os conjuntos de dados descartados descritos anteriormente, e em seguida avaliar com a ajuda de voluntários se a classificação fez sentido ou não, sendo assim uma tarefa não-supervisionada de aprendizado de máquina.

Referências

Barbosa, L., Pham, K., Silva, C., Vieira, M. R., and Freire, J. (2014). Structured open urban data: understanding the landscape. *Big data, v. 2, n. 3*, pages 144–154.

Colas, F. and Brazdil, P. (2006). Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer.

dos Santos Pinto, H. (2018). Alinhamento de categorias em portais de dados abertos com base em um subconjunto abrangente. Dissertação de Mestrado — Instituto de Computação, Universidade Federal Fluminense. Disponível em `http://www.ic.uff.br/PosGraduacao/frontend-tesesdissertacoes/download.php?id=898.pdf&tipo=trabalho`. Acessado em 2020-05-03.

dos Santos Pinto, H., Bernardini, F., and Viterbo, J. (2018). How cities categorize datasets in their open data portals: an exploratory analysis. *dg.o 2018: Proceedings of the 19th Annual International Conference on Digital Government Research*.

Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. C. P. L. F. D. (2011). *Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina*. LTC.

Frtunić Gligorijević, M., Bogdanovic, M., Veljkovic, N., and Stoimenov, L. (2019). Open data categorization based on formal concept analysis. *IEEE Transactions on Emerging Topics in Computing*, pages 1–1.

Jetzek, T., Avital, M., and Bjorn-Andersen, N. (2014). Data-driven innovation through open government data. *J. Theor. Appl. Electron. Commer. Res.*, 9(2):100–120.

Koesten, L. M., Kacprzak, E., Tennison, J. F., and Simperl, E. (2017). The trials and tribulations of working with structured data: -a study on information seeking behaviour. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1277–1289.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Infromation Retrieval*. Cambridge University Press.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

M.F.Porter (1980). An algorithm for suffix stripping. *Program, 14(3)*, pages 130–137.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press.

NLTK (2019). NLTK. Disponível em `https://nltk.org`. Acessado em 2019-11-25.

Open Knowledge International (20–a). The open definition. Disponível em `http://opendefinition.org/`. Acessado em 2019-11-25.

Open Knowledge International (20–b). Open knowledge international. Disponível em `https://okfn.org/`. Acessado em 2019-11-25.

Open Knowledge International (20–c). What is open? Disponível em `https://okfn.org/opendata/`. Acessado em 2019-11-25.

Porter, M. (2006). The porter stemming algorithm. `https://tartarus.org/martin/PorterStemmer/`. Acessado em: 2019-11-26.

Rajaraman, A. and Ullman, J. (2011). *Data Mining: Mining of Massive Datasets*. Cambridge University Press.

Reis, J. R., Viterbo, J., and Bernardini, F. (2018). A rationale for data governance as an approach to tackle recurrent drawbacks in open data portals. In *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, pages 1–9.

Sathya, R. and Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, 2(2):34–38.

Scikit-Learn (2019). Scikit-Learn. Disponível em `https://scikit-learn.org/`. Acessado em 2019-11-25.

Wang, J., Jiang, C., Zhang, H., Ren, Y., Chen, K.-C., and Hanzo, L. (2020). Thirty years of machine learning: The road to pareto-optimal wireless networks. *IEEE Communications Surveys & Tutorials*.

Xiao, F., He, D., Chi, Y., Jeng, W., and Tomer, C. (2019). Challenges and supports for accessing open government datasets: Data guide for better open data access and uses. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, CHIIR '19, page 313–317, New York, NY, USA. Association for Computing Machinery.

Yang, H.-C., Lin, C. S., and Yu, P.-H. (2015). Toward automatic assessment of the categorization structure of open data portals. In Wang, L., Uesugi, S., Ting, I.-H., Okuhara, K., and Wang, K., editors, *Multidisciplinary Social Networks Research*, pages 372–380, Berlin, Heidelberg. Springer Berlin Heidelberg.